Greenland vs Mars for Pre-Trained AI Compute: Nuclear Baseload, LEO Connectivity, and the Physics of Practicality

Douglas C. Youvan

doug@youvan.com

www.youvan.ai

January 6, 2026

The idea of putting large-scale AI compute “where it’s cold” sounds like a simple engineering hack: move the machines to the Arctic, let the air do the cooling, and run on clean energy. But the real question is not whether cold air helps—it does. The real question is where an AI facility can be powered, connected, repaired, and governed at the scale that modern models require. This paper develops a deliberately asymmetrical comparison between two extreme candidates: Greenland and Mars. Under an intentionally permissive thought experiment—Greenland becomes a U.S. state, nuclear regulation friction disappears, and LEO satellite connectivity is aggressively optimized—we still find that the deciding constraints are not political but physical. Latency, power density, logistics, and maintenance loops dominate the feasibility boundary. The argument is sharpened by focusing on pre-trained AI workloads—large inference fleets, distillation, evaluation, and fine-tuning—rather than frontier pretraining, which is bandwidth- and dataset-ingestion heavy. By reframing “data center siting” as an exercise in constraint geometry, the comparison becomes a template: it shows how to separate hype from feasibility, and how to identify the minimum set of infrastructure primitives required for any serious compute colony—on Earth or beyond.

Keywords: Greenland, Mars, AI data center, pre-trained AI, inference at scale, nuclear power, SMR, microreactor, Starlink, LEO satellite internet, latency, power density, Arctic infrastructure, remote operations, compute logistics, heat reuse, geopolitics of compute, resilience engineering, hyperscale constraints, sovereignty.

A collaboration with GPT-5.2-Thinking. 50 pages. CC4.0.

Outline

1. Problem Statement and Scope
1.1 Why “cold places for AI” became plausible
1.2 Why this paper compares Greenland to Mars
1.3 Assumptions of the thought experiment and what they buy us
1.4 Why we focus on pre-trained AI workloads, not frontier pretraining

2. The Constraint Geometry of AI Compute
2.1 Compute is not the bottleneck—power and logistics are
2.2 Latency as an operational, not merely user-experience, constraint
2.3 Bandwidth versus throughput: shipping models, not datasets
2.4 Reliability primitives: redundancy, repair loops, and spare parts

3. Greenland as a U.S. State: A Hypothetical “Compute Territory”
3.1 Site logic: ports, workforce, weather, and buildability
3.2 Cooling advantage: economization, liquid cooling, and failure modes
3.3 Heat as an output: district heating and local value capture
3.4 The role of local institutions: universities, research, and talent pipelines

4. Nuclear Baseload as the Enabler
4.1 Microreactors for pilots: fast deployment, modularity, and limits
4.2 SMRs for scale: campus growth from tens to hundreds of MW
4.3 Microgrid design: islanding, black start, and power quality for GPUs
4.4 Safety and governance in a “no friction” regulatory assumption

5. LEO Connectivity as the Control Plane and the Product
5.1 Starlink as bootstrap versus Starlink as backbone
5.2 Latency budgets for inference fleets, eval pipelines, and human oversight
5.3 When satellite fails: redundancy, routing, and degraded modes
5.4 The economic reality of bandwidth for AI services

6. Mars Compute: What Changes When Physics is the Policy
6.1 The tyranny of minutes: comm delay and autonomy-first operations
6.2 Power regimes: kilowatts, not megawatts, and what that implies
6.3 Thermal rejection without a thick atmosphere: radiators and dust
6.4 Maintenance and human labor under habitat constraints

7. Direct Comparison Matrix: Greenland vs Mars
7.1 Latency and operational tempo
7.2 Power scale, growth curve, and failure containment
7.3 Construction, replacement parts, and time-to-repair

7.4 Security and sovereignty: different threat models, different controls
7.5 Cost structure: capex, opex, and “distance tax”

8. Program Plan for a Greenland Pre-Trained AI Campus
8.1 Phase 0: 5–20 MW pilot inference site
8.2 Phase 1: 50–100 MW campus with mature remote ops
8.3 Phase 2: 200+ MW expansion and heat-reuse integration
8.4 Workload strategy: inference, distillation, fine-tuning, eval, archiving
8.5 Measuring success: uptime, cost per token, and resilience drills

9. What This Thought Experiment Teaches About Earth-Based Compute Futures
9.1 AI infrastructure as a new kind of strategic industrial base
9.2 “Compute colonies” on Earth: the minimum viable primitives
9.3 The boundary between engineering feasibility and narrative temptation
9.4 A reusable framework for siting decisions in extreme environments

10. Conclusion
10.1 Greenland is easier than Mars, and why that’s the wrong level of claim
10.2 The deeper result: feasibility follows constraint geometry
10.3 What would have to change for Mars to become competitive
10.4 Forward work: rigorous scenarios, benchmarks, and a disciplined Reference section

References
Art

1. Problem Statement and Scope

This paper addresses a deceptively simple question: if you wanted to place a large AI compute facility in an extreme environment, what actually determines feasibility? Popular discussion tends to fixate on one variable at a time—cooling, renewable energy, or "where it's cheap." But modern AI compute is a coupled system: power delivery, thermal management, networking, maintenance tempo, parts supply, governance, and human operations all interact. A site that "wins" on cooling can still fail on repair loops. A site that "wins" on power can still fail on latency. The goal here is not to argue that any particular location should be chosen in reality. The goal is to build a high-contrast comparison that reveals the true constraints and how they scale.

1.1 Why "cold places for AI" became plausible

"Put the data center where it's cold" used to sound like a gimmick. It became plausible when three trends converged.

First, the heat density of AI workloads jumped. General-purpose data centers grew up around relatively moderate rack densities and predictable thermal loads. AI clusters, especially with modern accelerators, concentrate power into smaller footprints and demand tighter thermal control. When heat density rises, cooling moves from a background utility to a first-order design constraint. Cold ambient air becomes not merely a cost reducer but an engineering simplifier: it expands the window for free cooling, reduces mechanical complexity, and buys operating headroom during unusual thermal events.

Second, power and cooling costs became inseparable from the business model. AI services convert electricity into tokens, embeddings, simulations, or evaluations. The economics are sensitive to marginal kilowatt-hours, cooling overhead, and uptime. In that environment, even modest efficiency gains matter, and a climate that consistently supports economization can translate into real competitive advantage.

Third, the geopolitics of compute intensified. Compute is no longer just a commercial input; it is increasingly treated as strategic infrastructure. That pressure pushes builders to diversify geography, reduce dependence on fragile supply corridors, and seek environments where resilience is an intrinsic property rather than a bolt-on expense. Cold regions entered the conversation not only as "cheap cooling" zones, but as potential anchors for a new class of infrastructure: compute that is power-secure, thermally stable, and operationally robust.

The result is a shift in intuition. "Cold" stopped being a curiosity and started looking like a structural advantage—provided the other constraints can be met.

1.2 Why this paper compares Greenland to Mars

Greenland and Mars are not competitors in any near-term market sense. They are chosen as conceptual extremes that clarify the feasibility boundary.

Greenland represents the harsh end of terrestrial reality: remote logistics, Arctic weather, limited local industrial base, and a constrained environment for construction and maintenance. Yet it remains on Earth: it allows shipping by sea, human travel, conventional repair practices, and integration into global networks. In other words, it tests what happens when you push Earth-based engineering into a demanding corner without leaving the planet.

Mars represents a different category entirely: an environment where physics is the policy. Distance imposes multi-minute communications delays. Resupply is episodic and expensive. Human labor is constrained by habitat requirements and risk. Even if you assume aggressive engineering capability, Mars forces you to confront what cannot be optimized away: speed-of-light latency, launch windows, and the operational asymmetry between "repairable" and "replaceable."

This comparison is therefore not about which place is "better." It is about what changes when you move from a remote terrestrial site to an off-world site. The contrast turns fuzzy arguments into crisp ones. On Earth, many bottlenecks can be mitigated by money, engineering, or policy alignment. On Mars, many bottlenecks are irreducible.

The paper's claim at this stage is intentionally modest: Greenland is far easier than Mars not because Greenland is easy, but because Mars is categorically different in the constraints it imposes. Greenland is a hard project. Mars is a different kind of project.

1.3 Assumptions of the thought experiment and what they buy us

The paper uses an intentionally permissive thought experiment to strip away debates that can distract from the true limiting factors. These assumptions are not endorsements. They are analytical scaffolding.

Assumption one: Greenland becomes a U.S. state. This removes an entire class of ambiguity around jurisdiction, long-term legal stability, and the alignment of major infrastructure policy with U.S.-scale financing and procurement norms. It lets us treat the site as "domestic critical infrastructure" rather than a cross-border venture that might be subject to shifting bilateral politics. What it buys us is not magic; it buys us administrative continuity and a simplified governance model for a large industrial build.

Assumption two: nuclear regulation friction is not a problem. In the real world, licensing, public acceptance, and safety oversight dominate timelines. Here, we temporarily suspend that friction to prevent the analysis from collapsing into a single bottleneck: permitting. The intent is to ask a different question: if baseload power is available on demand, what constraints remain?

This is crucial because even perfect permitting does not solve logistics, maintenance tempo, supply chains, workforce, and network architecture.

Assumption three: LEO satellite connectivity is adapted and optimized. This assumption clarifies the role of networking in two ways. First, it asks what happens if you treat satellite not as a last resort but as a primary control-plane and service-plane option for certain workloads. Second, it highlights a key distinction: low latency is helpful, but bandwidth and reliability are what determine whether the site behaves like a normal node on the internet or like a semi-isolated outpost. “Adapted” is not a handwave; it means engineered redundancy, dedicated ground infrastructure, and operational doctrine for degraded modes.

Assumption four: the facility runs pre-trained AI workloads. This narrows the problem from “build the biggest training factory” to “build a high-throughput compute colony that can operate with reduced dependence on constant data ingest.” This matters because it changes the bandwidth profile, the update cadence, and the operational definition of success.

Together these assumptions buy a cleaner result: if Greenland still looks dramatically easier than Mars under permissive assumptions, it will be because the remaining constraints are structural, not bureaucratic.

1.4 Why we focus on pre-trained AI workloads, not frontier pretraining

There are two distinct infrastructural animals that often get conflated: frontier pretraining and large-scale post-training deployment.

Frontier pretraining is a bandwidth- and dataset-ingestion intensive operation. It demands constant movement of massive corpora, frequent iteration, and a highly optimized training fabric. It benefits from proximity to deep network backbones, a mature supplier ecosystem, and rapid access to specialized personnel. It also tends to be sensitive to small perturbations: a minor network constraint, a parts delay, or a prolonged outage can cascade into expensive idle time across an entire cluster. In other words, pretraining amplifies operational friction.

Pre-trained AI workloads have a different geometry. Inference fleets, distillation pipelines, evaluation harnesses, fine-tuning for niche domains, and long-horizon batch processing can often be scheduled, queued, and buffered. They can tolerate higher variance in network throughput if the system is designed around caching and asynchronous transfer. They can be architected to “ship models, not datasets,” moving weights and deltas rather than continuously ingesting raw corpora. They can also benefit more directly from cold-climate thermals because the workload is sustained and predictable, and efficiency improvements translate into stable cost reduction.

Focusing on pre-trained workloads therefore serves two purposes. It makes the Greenland concept maximally plausible by aligning the workload with the strengths of a remote, power-

secure, cold environment. And it makes the Mars comparison maximally honest by showing that even when you choose the more Mars-friendly category of AI work—batchable, asynchronous, autonomy-compatible—the off-world constraint set still dominates.

This scope choice is also methodological. The paper is not trying to win by choosing an easy target. It is trying to separate what is intrinsically possible from what is narratively tempting. If even the most favorable AI workload class still maps Greenland into "hard but feasible" and Mars into "conceptually possible but structurally constrained," then we learn something durable about infrastructure in extreme environments.

In short: this is a paper about constraint geometry, not destination fantasy. Greenland is used to explore the upper edge of terrestrial feasibility. Mars is used to expose what changes when physics, not policy, becomes the limiting regulator.

2. The Constraint Geometry of AI Compute

"Constraint geometry" is a way of saying that AI infrastructure is not one problem but a coupled system of limits. You do not "build compute" in the abstract; you build a machine that converts electricity, cooling capacity, network transport, and human maintenance into sustained useful work. The output is not a peak benchmark number. The output is dependable throughput over time: tokens served, evaluations completed, jobs finished, incidents resolved, upgrades rolled out, and failures contained.

The key idea is that feasibility is shaped less by what is theoretically possible and more by which constraints bind first. Different sites bind on different limits. A dense urban metro might bind on land and grid interconnect. A remote Arctic location might bind on logistics and redundancy. An off-world site binds on time delay, resupply cadence, and the cost of every kilogram. When you compare Greenland to Mars, you are comparing two constraint landscapes. Greenland is harsh but remains inside Earth's industrial envelope. Mars is a different envelope entirely.

2.1 Compute is not the bottleneck—power and logistics are

It is easy to talk about GPUs and hard to talk about megawatts. But power is the first-order variable because it sets the ceiling on sustained compute. You can procure accelerators, racks, and fabrics. You cannot "procure" a stable, high-quality, expandable power feed without building a generation-and-delivery system. In practice, the compute you can run is proportional to the power you can reliably deliver, not the number of chips you can buy.

Power as a bottleneck has three layers.

The first layer is raw capacity. AI clusters are power-hungry and increasingly so as density rises. If a site cannot provide the next increment of power without years of buildout, the "scale plan"

becomes a narrative rather than a roadmap. This is why “nuclear baseload” is an enabling assumption in the Greenland thought experiment: it removes the typical constraint of limited local generation and converts expansion into an engineering schedule rather than a political negotiation. But even with nuclear “solved,” power remains limiting because generation is only the start.

The second layer is power quality and continuity. AI hardware is sensitive to voltage stability, harmonics, transients, and switching events. A facility can be “powered” and still be fragile if the power is noisy or if ride-through is inadequate. The true question is not “How many megawatts exist?” but “How many megawatts can be delivered with the stability and redundancy that keeps a GPU fleet running through faults?” This shifts attention to microgrid design, UPS strategy, fault isolation, and black-start capability. Even in a frictionless regulatory world, engineering discipline does not disappear.

The third layer is the logistics shadow cast by power. Power capacity implies physical infrastructure: generators, transformers, switchgear, cabling, cooling plants, and spares. These are bulky, heavy, and maintenance-intensive. The logistics bottleneck shows up in two forms: construction logistics and operational logistics.

Construction logistics includes ports, cranes, storage yards, road and haul constraints, seasonal weather windows, and specialized labor availability. A site can be thermally perfect and still be a construction trap if every critical component requires a heroic delivery effort.

Operational logistics is subtler. It is the difference between “a part exists somewhere in the world” and “a part is in the right place within the repair window.” AI fleets convert small delays into large opportunity costs because downtime scales with the number of idle accelerators. Logistics is therefore not an accessory; it is an input to uptime. A remote site must be designed with logistics as a primary variable: spare strategy, on-site inventory depth, modular replaceability, and maintenance tempo. On Earth, you can thicken this logistics layer with money and planning. On Mars, the logistics layer becomes destiny.

So the core claim of this subsection is simple: compute is the dependent variable. Power and logistics are the independent variables. The site that “wins” is the site where power can scale and logistics can keep pace.

2.2 Latency as an operational, not merely user-experience, constraint

Latency is often discussed as a user-facing number: how fast a chat response appears, how long a call takes to connect, whether streaming feels smooth. That framing is incomplete. For an AI data center, latency is also a constraint on operations—the tempo at which humans and systems can manage, troubleshoot, coordinate, and recover.

Operational latency matters in at least five ways.

First, it sets the speed of the control plane. Modern facilities are software-defined: orchestration, scheduling, telemetry, security policy, and incident response are mediated by networked systems. When latency is low, remote operations behave almost like on-site operations. When latency is high, operations shift toward autonomy, delayed confirmation, and "act now, learn later" patterns.

Second, latency constrains human-in-the-loop governance. If you want pre-trained models to be used safely—through evaluation gates, canary rollouts, policy checks, and rollback discipline—your oversight pipeline depends on fast feedback. The difference between milliseconds and minutes is the difference between interactive supervision and asynchronous audit. Greenland can plausibly run like a remote terrestrial facility with normal escalation procedures. Mars forces a fundamentally different governance style: you must pre-authorize behaviors because you cannot intervene quickly.

Third, latency shapes failure containment. Many incidents are time-sensitive: thermal excursions, cascading power events, network partitioning, or runaway workloads. Quick detection is not enough; you need quick action. With low latency, you can centralize expertise and push interventions from a distance. With high latency, local autonomy becomes mandatory, and the system must be designed to fail safely without external guidance.

Fourth, latency affects multi-site coordination. Even if pretraining is not the focus, many AI workflows are distributed: evaluation farms, model registries, artifact stores, and policy systems. High latency increases synchronization costs and pushes systems toward eventual consistency. That is not impossible, but it changes how you design deployments and how you interpret telemetry. Greenland can remain inside the familiar world of near-real-time coordination. Mars becomes an exercise in delay-tolerant computing, where operational reality looks more like deep-space mission control than like enterprise IT.

Fifth, latency is a psychological and organizational variable. Humans make decisions differently when feedback is delayed. Confidence, caution, escalation thresholds, and the willingness to intervene all shift when you know that every action is slow to verify. This is not a minor factor; it reshapes the entire operations culture.

This is why Greenland vs Mars is not a rhetorical comparison. It is a categorical one. Greenland can be integrated into the ordinary human tempo of infrastructure operations. Mars cannot. No amount of money removes that gap. The speed of light is the final regulator.

2.3 Bandwidth versus throughput: shipping models, not datasets

Bandwidth is a headline number. Throughput is the lived reality. In extreme environments, the difference becomes decisive.

Bandwidth is the theoretical pipe size. Throughput is the sustained delivered volume after protocol overhead, retransmissions, jitter, congestion, outages, and policy constraints. Satellite links, in particular, can advertise attractive peak rates while still delivering variable throughput depending on weather, contention, gateway routing, and network conditions. For a pre-trained AI facility, this distinction guides what kinds of workloads are sensible.

If you are doing frontier pretraining, you are continuously ingesting data, moving checkpoints, coordinating large distributed jobs, and pushing massive artifacts around. That demands high, steady throughput and predictable performance. It also demands a mature backbone and redundancy because training runs are sensitive to interruptions.

If you are running pre-trained AI workloads, you can redesign the traffic pattern. You can reduce dependence on constant bulk transfer by changing what you move.

The phrase “shipping models, not datasets” captures this shift. It has several concrete meanings.

One meaning is that the largest stable artifact is the model itself—weights, tokenizer, configuration, and evaluation harness. Once a model is staged locally, inference can run with minimal external dependencies. Updates can be shipped as deltas: fine-tuned adapters, quantized variants, distilled children, and parameter-efficient patches. This makes the network more like a supply line for periodic artifacts than like a firehose for continuous data.

A second meaning is that inputs can be compressed into “jobs” rather than “corpora.” Instead of pulling enormous raw datasets across the network, you ship curated task bundles: evaluation suites, benchmark packs, domain-specific fine-tuning sets, and policy test cases. These are smaller, more structured, and easier to cache and validate.

A third meaning is that physical logistics can substitute for network logistics. On Earth, it can be cheaper and faster to move a large dataset on physical media than to push it through constrained links. This is not romantic; it is simple arithmetic. Shipping becomes an acceptable method of bulk transfer when the alternative is slow, expensive, or unreliable throughput. Greenland can exploit this option. Mars cannot do it at the cadence required for modern AI practice.

A fourth meaning is architectural: you design your system to tolerate intermittent connectivity. You schedule inference and evaluation jobs locally, queue results, and synchronize when the link allows. This is not a concession; it is a design style. It turns the facility into a batch-capable compute colony rather than a constantly online training factory.

These shifts are precisely why your scenario—nuclear power plus adapted LEO connectivity—maps more naturally to pre-trained AI than to frontier pretraining. It is a workload-site match: stable baseload and thermals enable high utilization, while "good enough" connectivity supports artifact movement, control-plane telemetry, and product delivery without requiring continuous bulk ingest.

Mars, again, is different. The bandwidth story becomes secondary to the delay story and the resupply story. You can't compensate for minutes of latency by "caching harder" when human operations and failure response are structurally delayed. And you can't compensate for limited throughput when the cost of physical resupply is interplanetary.

2.4 Reliability primitives: redundancy, repair loops, and spare parts

Reliability is often treated as a property of equipment. In reality, it is a property of systems plus time. A facility is reliable when it can absorb component failure without losing service—and when it can restore full redundancy fast enough that the next failure does not cascade.

This is why reliability has primitives. They are not "best practices" so much as the irreducible building blocks of survivable operations.

The first primitive is redundancy, but redundancy must be specific. "Redundant power" means redundant at the right level: generation, feed, switchgear, distribution, and rack-level delivery. "Redundant network" means diverse physical paths and diverse failure modes, not two logical links that share the same vulnerability. "Redundant cooling" means redundancy that still works under the local environmental extremes, not redundancy that assumes mild conditions.

The second primitive is fault isolation. A system fails gracefully when a fault is bounded. This requires segmentation: electrical zones, network domains, and thermal containment that prevent local failures from becoming systemic.

The third primitive is the repair loop. Reliability is not only the ability to survive the first fault; it is the ability to return to a resilient state before the second fault. The key variable is mean time to repair in the real world, not in vendor documentation. Mean time to repair is dominated by human access, diagnostic clarity, parts availability, and the simplicity of replacement.

The fourth primitive is spare parts and inventory depth. A remote site must decide what it can afford to stock on-site and what it will pull "just in time." The more remote the site, the more "just in time" becomes "too late." The more expensive the downtime, the more inventory becomes rational. This is not merely a financial calculation; it is an existential one for extreme environments.

The fifth primitive is modularity. Modular design shortens repair loops by turning complex failures into simple swaps. If a failed component requires specialized rework rather than a replacement module, the repair loop lengthens and reliability collapses under repeated stress.

These primitives draw a sharp line between Greenland and Mars. Greenland can build thick redundancy and sustain repair loops because spares and people can arrive. The calendar is inconvenient, but it exists. Mars must be designed as if resupply is rare and intervention is delayed; therefore the on-site spare strategy and autonomy requirements explode. Reliability on Mars becomes less like data center engineering and more like spacecraft engineering, where failure modes must be pre-contained because repair is not guaranteed.

So the constraint geometry closes the loop: power and logistics set the scale ceiling; latency sets the operational tempo; throughput determines feasible workload types; reliability primitives determine whether the site behaves like a robust node or like a fragile outpost. Under your assumptions, Greenland can be made to sit within a terrestrial feasible region for pre-trained AI compute. Mars remains outside that region unless the entire technological basis of power, autonomy, and resupply is transformed.

3. Greenland as a U.S. State: A Hypothetical "Compute Territory"

Treating Greenland as a U.S. state collapses the usual "foreign megaproject" uncertainties into a single domestic infrastructure frame: long-horizon legal continuity, federal-scale procurement, standardized critical-infrastructure governance, and the ability to treat power, connectivity, and resilience as national capabilities rather than boutique experiments. In this thought experiment, Greenland becomes a new kind of territory—less like a conventional state competing for generic data centers and more like a purpose-built "compute frontier" where the climate and geography are used as first-class design inputs. The point is not to romanticize remoteness. The point is to show how a remote environment can be engineered into a stable, high-utilization compute colony—provided the primitives of power, logistics, and operations are designed correctly.

3.1 Site logic: ports, workforce, weather, and buildability

A serious AI facility is not placed on a map; it is placed on a supply chain. The first siting question is therefore not temperature. It is the ability to move mass. Transformers, switchgear, reactor modules, steel, piping, and prefabricated electrical rooms are not "cargo"—they are the project. A port with predictable access, heavy-lift capability, laydown space, and an all-season logistics plan is the enabling condition. The best site is rarely the coldest; it is the one with the most reliable path for heavy equipment and the fewest seasonal choke points.

Workforce is the second constraint. Even "lights-out" data centers are not people-free; they are people-minimized. You still need skilled technicians, electricians, controls engineers, safety staff,

network operators, and an escalation ladder for rare events. In Greenland, that means designing for a hybrid labor model: a local core team for routine operations and first response, plus a structured rotation of specialists for major maintenance, upgrades, and commissioning. The workforce strategy should be treated as an engineering artifact: training plans, certification paths, remote assist tooling, and a deliberate “bus factor” design so that no single person becomes a point of failure.

Weather and buildability together define the calendar. Arctic projects live and die by “construction windows” and “maintenance windows.” Wind, icing, storm seas, and low visibility affect shipping schedules, crane operations, and outdoor work. Buildability is therefore not a generic construction issue; it becomes part of the reliability story. The facility must be designed so that critical maintenance can be performed indoors as much as possible, and so that major components can be swapped without heroic weather-dependent procedures.

Finally, the siting logic must include the social build: housing, medical response, transport, and the mundane stability that keeps skilled workers willing to stay. In a compute territory, “community infrastructure” is not a philanthropic add-on. It is risk management, because high-reliability operations require a stable, rested, retained human system.

3.2 Cooling advantage: economization, liquid cooling, and failure modes

Cold climate is valuable because it expands the operating envelope, but it does not eliminate thermal engineering. It changes it. The Arctic advantage is not “no cooling needed.” The advantage is a broader range of hours in which the facility can reject heat with low mechanical complexity and lower overhead.

Economization—using outside air conditions to reduce or replace compressor-driven cooling—becomes a central design lever. In many climates, economization is intermittent; in Greenland it can be dominant. That changes the economic and reliability picture: fewer compressor hours can mean lower energy overhead and fewer mechanical wear-out failures. But economization in the Arctic introduces its own failure modes: ice ingestion, snow intrusion, damper freezing, sensor drift under thermal cycling, and brittle failures in materials not chosen for cold exposure. “Free cooling” is only free if the building envelope, filtration, and control system are designed to treat wind-driven snow and icing as normal operating conditions, not rare anomalies.

Liquid cooling becomes the other pillar, especially for dense AI racks. The core idea is straightforward: move heat from silicon to a fluid efficiently, then move that heat to a rejection system that can operate robustly in cold, variable weather. Liquid cooling also creates a path to heat reuse, which is strategically important in Greenland. But it introduces new operational disciplines: leak detection at scale, coolant quality control, corrosion management, pump redundancy, and procedures for safe service under low temperatures. A liquid-cooled facility

can be extraordinarily stable, but only if it is treated as a process plant rather than as an "IT room."

Failure modes must be designed explicitly because the Arctic punishes optimistic assumptions. The key thermal risks are not always "too hot." They can be "too cold in the wrong place," leading to condensation, frosting on coils, frozen pipes, and control instabilities. Temperature gradients can become more severe; thermal shock can crack or warp components. The correct design stance is to assume that every boundary condition will be hit—wind, cold snaps, icing storms, supply interruptions—and to build graceful degradation modes: reduced load operation, thermal buffering, and automated safe-state transitions that protect hardware even when external conditions become extreme.

In short: Greenland's climate can reduce thermal overhead and simplify steady-state operation, but it demands greater rigor in edge-case engineering. The cooling advantage is real, but it is earned.

3.3 Heat as an output: district heating and local value capture

In a conventional data center narrative, waste heat is a nuisance. In Greenland, heat is a scarce asset. That flips the framing: the facility is not only a compute plant; it is a heat plant with compute as the primary product and heat as a valuable byproduct.

Treating heat as an output changes design priorities. You begin to care about temperature grade and controllability. Low-grade heat can support district heating loops, building heat exchangers, greenhouses, aquaculture systems, or industrial preheating. Higher-grade heat—more feasible with certain liquid-cooling architectures—can expand the set of useful applications. The engineering implication is that you design the thermal system with a deliberate "export interface": heat exchangers, buffer tanks, metering, and control logic that can deliver stable heat even as compute loads vary.

Local value capture is not just goodwill; it is project stability. A compute territory succeeds when local stakeholders see tangible benefits that persist beyond construction jobs. Heat reuse can become the anchor benefit because it offsets imported heating fuels, reduces energy costs, and improves living conditions. It also creates a shared resilience story: the compute facility becomes part of the community's critical infrastructure, which can strengthen political support and justify higher reliability investments.

This is not automatic. Heat reuse requires coordination with municipalities, utilities, and building owners. It requires capital on both sides of the interface. It requires governance agreements about pricing, uptime expectations, and maintenance responsibilities. But if done well, it converts a remote AI facility from a "foreign object" into a local utility partner. In a harsh

environment, that integration can be the difference between a tolerated project and a beloved one.

3.4 The role of local institutions: universities, research, and talent pipelines

A compute territory cannot be sustained purely by imported expertise. It needs a local competence gradient: institutions that develop people, test ideas, and translate the facility into regional capability rather than a sealed industrial enclave.

Universities and research organizations play three distinct roles.

First, they are talent factories. Even if most advanced model work remains concentrated elsewhere, the daily reality of operating an AI campus demands skilled technicians, network staff, controls engineers, and safety personnel. Institutions can build curricula tailored to Arctic operations: power systems, industrial controls, cybersecurity for critical infrastructure, and data-center mechanical systems. The goal is not to turn every student into an AI researcher. The goal is to build a stable human substrate for high-uptime operations.

Second, they are local problem solvers. Greenland has domain priorities that pair naturally with compute: climate and ice modeling, fisheries and ecosystems, logistics optimization, language technology, and public-sector service modernization. A compute campus that offers defined "research compute allocations" can catalyze projects that matter locally and create positive feedback: local research use cases justify expansion, and expansion creates more capability.

Third, institutions provide legitimacy and continuity. A private-led industrial facility can be seen as transient, extractive, or opaque. Partnering with local institutions creates transparent channels for education, research, and public benefit. It also creates a pipeline of local leadership—people who understand the facility, can critique it, and can help govern it.

In this thought experiment, the "statehood" assumption strengthens these roles by enabling federal research funding channels, standardized education grants, and integration into national laboratory and university networks. That matters because it turns "remote campus training" into a long-horizon program rather than a one-off initiative.

The summary insight is that Greenland-as-compute-territory is not only a hardware story. It is a socio-technical story: ports and power enable the plant, but institutions and talent pipelines enable the decades-long operation that makes the plant more than a stunt.

4. Nuclear Baseload as the Enabler

If the central problem of AI infrastructure is sustained power at scale, then nuclear baseload is the most direct way to convert a remote site from "interesting" to "credible." It collapses two chronic constraints at once: the need for massive, continuous energy and the fragility of

depending on long supply chains for fuel. In the Greenland thought experiment, nuclear is not a cultural statement or a political argument. It is a design primitive: a way to guarantee that the compute colony has a stable energy floor independent of wind variability, seasonal hydrology, or imported fossil fuels.

But nuclear is not a single thing. It is a family of deployment modes. The paper separates microreactors and small modular reactors (SMRs) not because one is "better," but because they map to different phases of a compute program. Microreactors map to pilots and early scale when speed and modularity matter. SMRs map to sustained, industrial-scale growth where economics, maintainability, and grid-quality power delivery dominate.

4.1 Microreactors for pilots: fast deployment, modularity, and limits

A pilot AI campus has a paradox: it must be big enough to be meaningful but small enough to be buildable under uncertainty. The practical question is not "Can we get to 300 MW?" but "Can we establish a stable 5–20 MW class facility that proves the environment, logistics, and operations model?" Microreactors are attractive in this phase because they promise a deployment profile that looks more like "industrial module installation" than like "major power plant construction."

The core advantages for pilots are speed, modularity, and separability.

Speed means you can anchor the first site without waiting for a multi-year regional power build-out. If the goal is to validate a compute territory, time-to-first-power is the decisive milestone. A microreactor-based pilot can be framed as a testable unit: it delivers a defined power envelope, requires a defined maintenance cadence, and can be instrumented heavily to learn operational truths quickly.

Modularity means you can scale by adding units rather than by overbuilding one giant plant from day one. This is especially valuable in a place where construction windows are constrained and where the unknowns are not only technical but logistical and human. A modular approach aligns with the principle of reversible commitment: build a meaningful first step, learn, then expand.

Separability means the pilot can be designed as an islandable microgrid from the start. That reduces dependence on any local grid limitations and allows the campus to behave like a self-contained power island—an operational style that is closer to how the site will ultimately need to function if it becomes a true compute territory.

But microreactors also have hard limits.

The most obvious is scale. A microreactor that produces a few megawatts is transformative for a remote community or a small industrial facility, but a modern AI campus quickly grows beyond

that envelope. You can parallelize units, but at some point the complexity of many small units starts to look worse than the simplicity of a well-designed medium-scale plant. There is a “too many boxes” problem: each unit implies its own maintenance, monitoring, safety systems, and spares strategy.

The second limit is operational maturity. A pilot built around a microreactor is, by definition, operating closer to the frontier of nuclear productization. Even in your “no friction” regulatory scenario, the reality of operating an early-generation system in a harsh environment is that you will encounter lessons that are invisible on paper: how maintenance actually works in cold weather, how supply chains behave, what the failure patterns look like, and how human teams interact with the system.

The third limit is economic. Microreactors may be economically attractive for remote locations that would otherwise burn expensive imported fuel, but cost per megawatt can be high relative to larger plants. For the pilot phase, that may be acceptable because the goal is not lowest cost—it is feasibility demonstration. But for long-term scale, economics becomes a dominant constraint.

So microreactors are best understood as the “fast start” tool: a way to establish proof of operations, not necessarily the endpoint architecture.

4.2 SMRs for scale: campus growth from tens to hundreds of MW

If microreactors make the pilot plausible, SMRs make the growth curve plausible. Once the campus becomes large enough that outages have enormous opportunity costs and that operations must be industrial rather than experimental, you need a power plant architecture that is designed for routine maintainability, predictable performance, and credible scaling.

SMRs align with the campus growth problem in three ways.

First, they support stepwise expansion at meaningful increments. A data center campus rarely grows smoothly; it grows in blocks. You commission a new hall, a new substation bay, a new cooling module, and you fill it over time. SMRs allow power expansion to occur in similarly discrete steps: add a module, integrate it into the microgrid, increase the base load, then add the next compute block.

Second, SMRs improve the “system count” problem. Instead of many microreactors, you can run fewer larger units with fewer duplicated subsystems, fewer maintenance trains, and a simpler spare-parts regime. This matters in a remote environment where maintenance time and inventory management are structural costs.

Third, SMRs map to “utility-grade” power quality. AI campuses do not just want power; they want stable power that behaves predictably under changing loads. As the campus grows, the

power plant must interact gracefully with the rest of the electrical system: redundancy, load shedding, reactive power management, and fast response to transients.

In the Greenland thought experiment, the SMR phase is where the compute territory concept becomes more than a pilot. It becomes a strategic industrial base: a site that can host large inference fleets, run continual evaluation and distillation, and serve as a stable export point for AI services—without being at the mercy of a limited local grid.

An important nuance: scaling from tens to hundreds of megawatts is not only a generation problem. It is a delivery problem. As power rises, electrical infrastructure must scale with it: transmission within the campus, substations, distribution redundancy, and fault isolation. The SMR is the heart, but the arteries determine survivability.

4.3 Microgrid design: islanding, black start, and power quality for GPUs

A Greenland compute territory would be a microgrid by necessity, not by fashion. The microgrid frame is essential because it turns the campus into a controllable electrical island whose behavior can be engineered and tested. In a remote environment, this is the difference between "a site that has power" and "a site that can keep power through faults."

Islanding is the ability to operate independently of any external grid disturbances. Even if you connect the campus to local infrastructure, you do not want local faults to propagate into your facility, nor do you want your facility's load dynamics to destabilize local systems. Islanding creates decoupling. It also forces discipline: you must own frequency control, voltage regulation, and load balancing. The campus becomes a small utility, and it must behave like one.

Black start is the ability to restart after a total outage without relying on external power. This is not a theoretical requirement. In a harsh environment, you must assume that extreme events can produce total system loss: a cascading fault, a protection misfire, a storm-related physical failure. A campus that cannot black start is a campus that can be trapped in downtime while waiting for external restoration. A black-start-capable design includes a hierarchy of startup resources: auxiliary generators or batteries, controls and communications that can boot without the main plant, staged energization of switchgear, and a controlled ramp of compute load.

Power quality is the quiet killer. GPUs and high-density power supplies create and respond to electrical conditions in ways that can stress a weak system. Large clusters ramp load quickly; they can generate harmonics; they can expose weaknesses in protection schemes. The campus must therefore treat power conditioning as a first-class system: harmonic filtering, reactive power control, fast-switching UPS architecture, and protection coordination designed for high-density electronic loads. You do not want a power system that is "within spec on average." You want a system that remains stable under disturbances, because those disturbances are when the money is lost.

A critical microgrid design principle for AI is graceful degradation. When the system is stressed, you should be able to shed load in a controlled way, preserving core services and avoiding hardware damage. This requires intimate integration between electrical systems and the compute scheduler. The facility must be able to say: “We are in a constrained power mode; reduce nonessential jobs; preserve critical inference lanes; keep thermal stability; maintain control-plane integrity.” This is not a future fantasy. It is the operational style that makes remote compute sustainable.

4.4 Safety and governance in a “no friction” regulatory assumption

The “no friction” assumption is analytical, not ethical. It does not mean “ignore safety.” It means “do not let licensing timelines dominate the argument.” Even with regulation friction removed, safety and governance remain central because they are the difference between an operational asset and a fragile liability.

In practice, nuclear governance in a compute territory would be shaped by four needs: containment of failure, public trust, operational transparency, and integration with critical infrastructure doctrine.

Containment of failure means designing systems so that faults—whether technical or human—do not cascade into unacceptable outcomes. That is a design problem and a training problem. It implies redundant safety systems, conservative operating margins, and rigorous maintenance discipline.

Public trust is not automatic in any nuclear project, and in a remote region it can be even more sensitive. In a statehood scenario, the project must be legible to the public: clear safety commitments, transparent reporting, independent oversight, and demonstrable preparedness. “No friction” cannot mean “no legitimacy.” If legitimacy collapses, the project becomes politically unstable, which is a different kind of friction that no engineering can fully overcome.

Operational transparency means instrumentation and auditability. A compute territory that runs nuclear baseload would need a culture of measured reality: continuous monitoring, incident logging, and post-event analysis that is shared with the relevant oversight bodies. This is true even in a permissive scenario because it protects the project from complacency—the single most dangerous psychological failure mode in high-reliability operations.

Integration with critical infrastructure doctrine means treating the nuclear plant and the compute campus as a unified system from a security perspective. Cybersecurity is not an add-on when industrial controls meet high-value compute. Physical security is not an aesthetic concern when the site becomes strategic. The governance posture must therefore assume that the facility will be a target—of espionage, sabotage, or political disruption—and must be designed to remain safe and recoverable under adversarial conditions.

The deeper point of this section is that nuclear baseload is enabling precisely because it moves the facility out of “power scarcity mode” and into “operational discipline mode.” Once power is abundant and stable, the success of the compute territory depends on how well it is run: how quickly it can repair, how gracefully it can degrade, how transparently it can govern itself, and how deeply it can integrate into the social fabric of the region.

Nuclear, in this paper, is therefore not the conclusion. It is the enabler that makes the real problem visible.

5. LEO Connectivity as the Control Plane and the Product

A remote AI campus succeeds only if it is legible to the rest of the world: schedulers must dispatch work to it, security teams must monitor it, engineers must maintain it, and customers must reach it. In most data center narratives, connectivity is treated as plumbing—something you buy in bulk and then forget. In a “compute territory” narrative, connectivity becomes architecture. It divides into two planes.

The first is the control plane: the nervous system that runs the campus—telemetry, orchestration, configuration management, security policy, incident response, and human remote operations. The control plane must remain available in degraded conditions because it is what lets you stabilize and recover.

The second is the service plane: the data movement and product delivery path—model artifacts, evaluation results, customer queries, and the outward-facing interfaces that make compute valuable. The service plane is where bandwidth and throughput matter.

LEO satellite internet is interesting here because it changes the usual trade: higher latency than fiber but far lower latency than traditional geostationary satellite links, with wide-area coverage and fast deployability. In your scenario, LEO connectivity is “adapted” and optimized. That means we can treat it as a serious tool—not merely a temporary patch—while still acknowledging that satellite physics and economics impose ceilings that fiber does not.

5.1 Starlink as bootstrap versus Starlink as backbone

There are two different uses of Starlink-like LEO systems in a Greenland compute territory, and confusing them creates bad design decisions.

As bootstrap, LEO is a time-to-first-capability accelerator. You can start the project without waiting for terrestrial fiber upgrades, complex permitting for new undersea routes, or large-scale terrestrial backhaul construction. Bootstrap connectivity supports early construction operations, commissioning, remote engineering support, and the initial pilot compute service. It

also provides a path to “first customer traffic” and “first telemetry,” which are psychologically and politically powerful milestones.

Bootstrap LEO has three virtues: speed, coverage, and independence from local terrestrial infrastructure. It can work from day one, it can cover remote sites, and it can operate even when local networks are limited. For pilots and early phases, those virtues are decisive.

As backbone, the story changes. Backbone implies that the majority of production traffic—especially for a large campus—moves over LEO links. That can be plausible for certain product profiles and campus designs, but it demands explicit choices.

A backbone role is plausible when the product is compute-local and the external traffic is relatively light compared to the on-site work. That is exactly why focusing on pre-trained AI workloads matters. If the campus mainly runs inference and evaluation with staged models and cached job bundles, you can treat LEO as a viable backbone for the service plane—because your biggest data flows are internal: GPU-to-GPU, storage-to-GPU, intra-campus artifact movement. External traffic becomes dominated by request/response and periodic artifact sync, not continuous bulk ingest.

Backbone LEO becomes fragile when the product requires sustained high-throughput bulk transfer—frontier pretraining, constant dataset ingest, large multi-site synchronous training, or any workflow that treats the network as a high-availability firehose. In that regime, LEO is better as redundancy than as primary.

In a properly designed compute territory, the intended evolution is therefore clear: bootstrap with LEO, prove operations, then add fiber to raise the throughput ceiling and reduce reliance on a single connectivity modality. But your scenario allows a different, still coherent path: remain primarily LEO-backed by choosing products that are compatible with that ceiling, while using physical shipping and asynchronous design to manage bulk artifacts.

The key insight is that “Starlink as backbone” is not a claim about satellite superiority; it is a claim about workload shaping. You make satellite work by making your service pattern satellite-friendly.

5.2 Latency budgets for inference fleets, eval pipelines, and human oversight

Most discussions of LEO latency stop at “it’s low enough for video calls.” That is true but not the point. For an AI campus, latency budgets are design constraints on workflows, oversight, and failure response.

Inference fleets have a relatively simple latency profile: the user or upstream service sends a prompt, the model responds. The total experience depends on both network latency and model generation time. For many pre-trained inference workloads, model generation time dominates;

the extra tens of milliseconds from LEO versus fiber is not decisive. This is especially true for long responses, high-context tasks, and batch inference where requests can be queued and responses returned asynchronously. LEO can therefore support a meaningful fraction of inference product traffic, especially if the system is designed to tolerate jitter and occasional slowdowns.

Evaluation pipelines have a different latency budget. Many eval tasks are batch by nature: run a suite, produce metrics, store artifacts, and report results. These pipelines are naturally compatible with asynchronous networking. The important requirement is not low latency but reliable delivery and integrity: results must arrive, artifacts must be verifiable, and the pipeline must be robust to temporary connectivity gaps.

Human oversight has the strictest operational latency budget—not because humans need millisecond response times, but because human intervention has a time value in incidents. Consider a failure that requires an operator to diagnose, decide, and act. If the control loop is interactive, a remote expert can stabilize the system quickly. If the control loop is delayed or intermittent, the system must operate autonomously and the human becomes an auditor rather than an operator.

In Greenland, LEO latency is “interactive enough” to preserve terrestrial operations culture. Engineers can SSH, monitor, deploy, and intervene with normal tempos. The relevant design question becomes: how do you protect that control plane so it remains available during partial outages? The answer is to prioritize the control plane, reserve bandwidth, and design multiple independent paths for control traffic.

The moment you compare to Mars, the meaning of latency changes entirely. Multi-minute one-way delays force autonomy-first operations. Human oversight becomes fundamentally asynchronous. The Greenland campus can still have human-in-the-loop gating and rollback in near-real-time. Mars cannot. That is a categorical difference in operational posture, not a performance tuning detail.

So the practical message is: if LEO is “good enough” to keep the control plane interactive, Greenland can be operated like a normal remote terrestrial site. The control-plane success criterion is not raw throughput; it is guaranteed availability under stress.

5.3 When satellite fails: redundancy, routing, and degraded modes

Satellite links fail in ways that are different from fiber. Fiber fails catastrophically: a cut can drop a route entirely until repaired. Satellite can fail partially: local weather impacts, congestion, gateway routing changes, beam handovers, terminal failures, and regional outages can degrade throughput and increase latency without an obvious single “break.” These failure modes demand a different resilience doctrine.

A compute territory must assume that connectivity will degrade, and design “degraded modes” that preserve safety, integrity, and partial usefulness.

The first principle is control-plane priority. The campus must remain controllable even when the service plane is impaired. That means out-of-band management, reserved bandwidth, and strict traffic shaping. You do not let bulk transfers starve monitoring, alerting, and remote access. In degraded mode, you shed service throughput before you lose observability.

The second principle is path diversity. Even if you assume LEO is optimized, you still want at least two independent connectivity modalities. In your scenario, “Starlink adapted” can include multiple gateways, multiple terminal clusters, and multi-constellation backup. But it is still wise to treat any single constellation as a common-mode risk: if there is a systemic failure, you lose the entire modality. The resilience posture therefore favors multi-path strategies—LEO plus any available terrestrial backhaul, plus an alternate satellite option, plus local autonomy.

The third principle is routing discipline and caching. In a satellite-first design, you must assume variable throughput. That pushes you toward store-and-forward patterns: local artifact caches, local package mirrors, local model registries, and local evaluation datasets. The campus becomes “network tolerant” by design. When connectivity returns, the system syncs deltas, not full states.

The fourth principle is job control under connectivity stress. If the campus loses service-plane throughput, it should not collapse. It should transition to local batch operation: keep running queued inference and evaluation tasks that do not require immediate external inputs, store results locally, and re-sync later. This is where pre-trained workloads shine. Frontier pretraining would suffer because it is tightly coupled to continuous coordination and artifact movement. Inference and eval can keep producing value offline and export it later.

The fifth principle is explicit operational doctrine. Remote sites fail when engineers assume the network will behave like fiber. A compute territory needs playbooks: “satellite degraded,” “satellite down,” “gateway instability,” “control-plane only,” “total isolation.” These are not just procedures; they are design targets. You architect the system to have meaningful behaviors in each mode.

This resilience design is where Greenland and Mars diverge again. Greenland can design degraded modes as exceptions. Mars must design degraded mode as the default condition. That alone changes the entire software and operational stack.

5.4 The economic reality of bandwidth for AI services

Even if LEO works technically, there is a second constraint: cost. Bandwidth is not just a physics problem; it is an economic one. The profitability of AI services is sensitive to the cost of electricity, the cost of compute, and the cost of moving bits.

For many AI services, especially those dominated by compute time, bandwidth is a small fraction of the total cost. A text response is not large. A sequence of prompts and completions is modest. If the campus serves inference at scale, the primary costs are power, capex amortization, and hardware utilization. In that regime, paying a premium for connectivity can still be rational because it enables remote siting and power advantages.

But bandwidth becomes expensive when the product involves large payloads: high-resolution generative media, massive embeddings for bulk corpora, large model checkpoint transfers, or continuous dataset ingest. In those cases, the cost and variability of satellite throughput can become a bottleneck even if the raw connection "works." The unit economics can flip. The site can become profitable for some product lines and unprofitable for others.

This is why "shipping models, not datasets" is not just an engineering slogan. It is an economic strategy. If you reduce external transfer volumes by caching, compressing, staging, and moving only deltas, you reduce recurring bandwidth spend. If you make the campus product primarily compute-heavy and bandwidth-light, you align the business with satellite realities.

A Greenland compute territory under your assumptions can therefore be economically coherent if it follows a disciplined product strategy:

Build around inference, evaluation, distillation, and fine-tuning artifacts that are compact relative to compute time. Avoid product lines that require massive continuous external data flows unless and until terrestrial fiber is added. Use physical shipping for bulk dataset updates when needed. Treat satellite as the control plane and as a moderate-throughput service plane, not as an infinite pipe.

Mars fails this economic test even more sharply. The bandwidth cost is not only monetary; it is temporal and operational. When the link has multi-minute delay and constrained throughput, "bandwidth as a product channel" becomes a mission architecture problem, not a pricing problem. You can do some AI work on Mars, but the economics and the operational reality look nothing like a normal service business. They look like a research outpost.

So the economic conclusion is aligned with the constraint geometry: LEO is viable when it is treated as a control-plane guarantee and a bandwidth-limited service plane, and when the product is designed to be compute-dominant rather than transfer-dominant. Greenland can be shaped into that mode. Mars demands a different civilization-scale infrastructure story before similar economics can even be imagined.

6. Mars Compute: What Changes When Physics is the Policy

Mars is not “a colder Greenland.” It is not even “a harder Antarctica.” It is an environment where many of the assumptions that make terrestrial infrastructure feel normal simply do not hold. On Earth, you can usually buy your way out of constraints: add redundancy, fly in experts, ship in parts, build a second route, upgrade the plant. On Mars, the constraints are not primarily contractual or logistical. They are physical and temporal. The planet itself imposes a governance regime: long delays, scarce power, hostile external conditions, and repair loops measured in mission timelines.

This section treats Mars compute as an infrastructure category, not a science-fiction backdrop. The question is not whether you can run computation on Mars—you can. The question is what kind of computation can be sustained, at what scale, with what operational posture, and with what definition of “success.” When physics is the policy, the default mode is not “high throughput with occasional incidents.” The default mode is “autonomous survival with occasional connectivity.”

6.1 The tyranny of minutes: comm delay and autonomy-first operations

On Earth, even remote operations rely on a shared human tempo. A technician can call an expert. An engineer can log in, run diagnostics, and push a fix. An incident commander can coordinate response in near-real time. That entire pattern collapses when the communications loop stretches into minutes.

The first consequence is that the control plane becomes asynchronous. You can still send commands and receive telemetry, but you cannot treat the loop as interactive. By the time you observe a problem and respond, the system may already be in a different state. This changes incident response from “diagnose and intervene” to “predict, pre-authorize, and constrain.” The system must decide locally how to stabilize itself, because waiting for human direction is equivalent to doing nothing.

The second consequence is that autonomy is not a feature; it is the operating system. A Mars facility must run with high local decision authority: fault detection, isolation, and recovery must happen on-site. Work scheduling must be robust to isolation. Security posture must be resilient without constant external verification. Even routine operations—software updates, configuration changes, maintenance planning—must assume delayed confirmation and slower rollback.

The third consequence is governance by precommitment. If you want oversight, you cannot “watch and stop” in real time. You must encode policies that the system enforces locally, and you must design for safe-state behaviors when the system encounters ambiguity. This is true for

compute in general and especially true for AI compute, where workload behavior can be complex and failures can cascade quickly.

The deeper implication is cultural: Mars operations resemble spacecraft operations more than data center operations. On Earth, you can run a facility with centralized expertise and remote intervention. On Mars, you must run a facility with local competence, local autonomy, and pre-planned constraint envelopes. Minutes of delay are not just an inconvenience; they define what "operations" means.

6.2 Power regimes: kilowatts, not megawatts, and what that implies

Modern AI infrastructure is built around sustained high power. The compute you can run is anchored in the power you can deliver continuously with quality and redundancy. A terrestrial AI campus grows in the language of tens to hundreds of megawatts because that is the scale at which large accelerator fleets are economically meaningful.

Mars power looks nothing like that today. The plausible near-term regime is constrained power: kilowatts to perhaps modestly larger, with tight budgets, high premium on efficiency, and strong coupling between power availability and mission safety. This single fact forces a radical shift in what "AI data center" could even mean.

The implication is not merely "smaller." It is "different."

First, you cannot treat compute as a commodity resource. On Earth, you can waste some power and still be profitable. On Mars, every watt competes with life support, habitat heating, mobility, communications, and scientific instrumentation. Compute is no longer just a business decision; it is a mission decision. Any nonessential workload must justify itself against survival-critical loads.

Second, utilization strategy changes. On Earth, the goal is high utilization because idle hardware is expensive. On Mars, the goal is alignment with variable power availability and system health. You may run opportunistic compute when power is abundant and shut down aggressively when it is scarce, because continuity of life-support and core systems is non-negotiable.

Third, hardware architecture changes. Earth-scale AI assumes dense accelerator clusters, high-throughput fabrics, and large storage systems. Mars-scale compute favors extreme efficiency: lower-power accelerators, aggressive quantization, smaller models, and a bias toward local inference and decision support rather than global service throughput. The more the power regime is constrained, the more the "AI workload" becomes embedded and mission-specific rather than "general AI services."

Fourth, growth becomes discontinuous. On Earth, you can scale by adding more power and more halls. On Mars, scaling is gated by major changes in energy infrastructure. You don't "add a module" casually. You redesign the settlement's power economy.

This is why Greenland's nuclear baseload assumption is transformative: it moves the problem into the terrestrial regime where megawatt-scale thinking is normal. Mars keeps you in a regime where compute is rationed, and the infrastructure story is closer to survival systems than to industrial campuses.

6.3 Thermal rejection without a thick atmosphere: radiators and dust

Mars can be cold, but cold does not automatically mean easy cooling. On Earth, much of data center cooling relies on moving heat into air and then rejecting it through a combination of convection and evaporation, assisted by fans, heat exchangers, and ambient conditions. A thin atmosphere changes the rules. You cannot assume the same convective efficiency, and you cannot rely on "free cooling" as if it were simply opening a vent.

Thermal rejection becomes an engineered radiator problem. Instead of dumping heat into thick air, you design surfaces and systems to move heat out reliably under conditions that include extreme thermal cycling. This is a different failure landscape from terrestrial HVAC.

Dust is the second dominant thermal enemy. Mars is famous for dust, and dust is a system-wide adversary: it affects moving parts, seals, radiative surfaces, solar panels, sensors, and filtration. A radiator covered in dust rejects less heat. A heat exchanger with fouled surfaces loses performance. A system that depends on steady airflow can degrade unpredictably if filters clog or if intake paths become compromised.

Thermal cycling and material stress become structural risks. Repeated temperature swings and harsh environmental exposure can stress joints, seals, and interfaces. Even if the internal compute modules are protected, the heat rejection path must interface with the outside world. That interface is where reliability is won or lost.

In terrestrial Greenland, the cold can simplify steady-state cooling if you engineer for icing and wind-driven snow. But the core cooling mechanisms remain familiar: air is dense, convective exchange is effective, and repair is accessible. On Mars, the rejection problem is intrinsically more brittle and more dependent on correct design from the start—because repair loops are slow and failure margins are thin.

So the thermal lesson is: Mars is not "naturally cooled." Mars is "thermally engineered." The cold is an input, not a solution.

6.4 Maintenance and human labor under habitat constraints

On Earth, maintenance is an assumption: you can send people to the site, work outdoors, bring in equipment, swap parts, and recover from mistakes. Remote locations complicate this, but the baseline remains intact. On Mars, the baseline is gone. Human labor is constrained by habitat life support, exposure risk, and limited time outside controlled environments. Maintenance is not just harder; it becomes a primary driver of system design.

The first consequence is that replaceability must be designed into everything. You want modules that can be swapped quickly with minimal tools, minimal exposure time, and minimal diagnostic ambiguity. Field repair is dangerous and time-consuming. The system must be modular not as a convenience but as a survival property.

The second consequence is that spares strategy becomes civilization-level planning. On Earth, you can stock spares and replenish them. On Mars, you must anticipate failure modes far in advance, because replenishment is slow and episodic. This favors conservative, redundant design and discourages fragile complexity.

The third consequence is that robotics and internalized maintenance become central. If you can move maintenance tasks into controlled environments, you reduce risk. That means design choices like indoor service corridors, sealed thermal loops, and remote manipulators. It also means the system must be introspective: rich telemetry, self-diagnosis, and predictable failure progression, because guessing wrong wastes scarce human effort.

The fourth consequence is operational psychology. In Earth data centers, a mistake is expensive. On Mars, a mistake can be mission-threatening. That changes culture: slower change management, heavier validation, stricter rollback discipline, and a bias toward stability over novelty. This clashes with the fast iteration culture common in terrestrial AI deployment. On Mars, “move fast and break things” is not a slogan; it is a fatal error mode.

So the maintenance conclusion is stark: a Mars compute facility must be designed like a life-critical spacecraft subsystem, because it is functionally entangled with life support and settlement continuity. Greenland, even as a harsh compute territory, remains within Earth’s repairability envelope. You can fly in parts. You can rotate specialists. You can do major repairs with conventional tooling. That is the difference between “difficult” and “categorically different.”

Taken together, these four subsections explain why Greenland is far easier than Mars even under optimistic assumptions. Greenland is a remote industrial project whose constraints can be thickened and managed. Mars is an environment where constraints define the architecture and compress the feasible design space. On Earth, policy and money can widen the envelope. On Mars, physics is the envelope.

7. Direct Comparison Matrix: Greenland vs Mars

A comparison matrix is useful only if it avoids the usual trick: picking one favorable dimension and pretending it dominates all others. Here we do the opposite. We compare Greenland and Mars across the variables that actually govern AI infrastructure over time: operational tempo, power regime and growth, repair loops, security posture, and the "distance tax" that turns small uncertainties into large costs. The purpose of this matrix is not to crown a winner. The purpose is to show why the two domains live in different feasibility classes even under optimistic assumptions.

Greenland is harsh but terrestrial: ships, ports, human travel, familiar physics, and repairable systems. Mars is harsh and extra-terrestrial: minutes of delay, episodic resupply, constrained human labor, and a power economy closer to survival systems than to industrial baseload. Under your thought experiment, Greenland inherits U.S.-state governance, frictionless nuclear deployment, and optimized LEO connectivity. Mars does not inherit anything; it inherits physics.

7.1 Latency and operational tempo

Operational tempo is the speed at which a facility can sense, decide, act, and confirm—across both human and automated control loops.

In Greenland, even if connectivity is satellite-first, latency remains in the human-interactive regime. Engineers can operate the site remotely with near-normal tooling: live dashboards, rapid escalation, interactive debugging, and tightly looped rollout/rollback cycles. The campus can maintain a "centralized expertise" model: a small local team handles hands-on tasks, while deep expertise can reside elsewhere and still intervene in near-real time. The operational culture can remain recognizably terrestrial: incidents are handled with fast collaboration and fast feedback.

On Mars, the delay forces an autonomy-first tempo. Human oversight becomes asynchronous and precommitted. The system must contain faults locally, choose safe states without waiting for instruction, and run for extended periods with limited external interaction. That changes what "operations" means: incident response becomes design-time policy rather than runtime coordination. Even routine configuration drift and software updates must be engineered as slow, heavily validated procedures. The operational tempo is defined by minutes of latency and by the reality that "confirmation" is delayed. In effect, Mars runs like a deep-space mission control problem, not like a remote data center.

This is the most decisive categorical divider because it is irreducible. You can spend money to improve Greenland's link redundancy and throughput. You cannot spend money to make Mars interactive.

7.2 Power scale, growth curve, and failure containment

Power is the master variable for AI compute because it determines sustained throughput, dictates thermal design, and sets the scale of the supporting electrical infrastructure.

In Greenland under the assumed conditions, power can be engineered into an industrial regime. With nuclear baseload treated as deployable, the campus can plan growth in familiar steps: a pilot that proves operations, a first major expansion into tens of megawatts, then further expansion into hundreds of megawatts. The growth curve can be designed as a sequence of discrete power blocks, each matched to new compute halls and new redundancy layers. Failure containment becomes a microgrid engineering problem: islanding, segmented distribution, reserve margins, and graceful load shedding that preserves control plane integrity.

On Mars, power lives in a scarcity regime. The settlement's power budget must cover life support, habitat heating, mobility, communications, and only then compute. Growth is therefore not a smooth campus plan; it is a settlement-level transformation problem. Failure containment is also harsher: there is less slack, fewer backup options, and less tolerance for cascading faults. If power fails, the compute problem is subordinate to survival-critical loads. On Earth, a data center outage is a business incident. On Mars, a power incident is existential.

The qualitative difference is not "Greenland has more power." It is that Greenland can treat power as an expandable industrial input, while Mars must treat power as a rationed life-critical resource.

7.3 Construction, replacement parts, and time-to-repair

Every infrastructure plan eventually becomes a logistics plan. In extreme environments, logistics is the hidden variable that determines whether uptime is real or aspirational.

Greenland is hard but tractable. Construction can use familiar methods: prefabricated modules, seasonal shipping, heavy-lift operations, rotating specialist crews, and conventional toolchains. Replacement parts can be stocked on-site and replenished by sea or air. The repair loop is measured in days to weeks for many failures, and the system can be designed for modular swaps that keep mean time to repair within tolerable bounds. Time-to-repair is still a binding constraint—remote sites must overstock spares and design for maintainability—but it remains within the human time scales that modern enterprises are built around.

Mars breaks that model. Construction is constrained by payload mass, launch schedules, and the need to assemble systems under habitat constraints with limited human EVA time. Replacement parts are not replenished casually; they are planned far in advance and delivered episodically. The repair loop can be measured in months, not days, and must be designed with the assumption that some repairs will not be possible. This forces a different design philosophy: extreme modularity, high redundancy, and conservative operating margins. It also forces a

different operational culture: fewer changes, slower upgrades, and a bias toward stability. The facility becomes more like a spacecraft subsystem than like an IT asset.

So the direct comparison is stark: Greenland's systems remain fundamentally repairable. Mars systems must be survivable even when not repairable on the desired timeline.

7.4 Security and sovereignty: different threat models, different controls

Security is where many superficial comparisons fail. People assume Mars is "safer" because it is remote, or Greenland is "riskier" because it is geopolitically relevant. The truth is that both are security-intensive—but the threat models are different.

In Greenland, the threat model is terrestrial and mixed. The campus would be a high-value strategic asset: vulnerable to espionage, cyber intrusion, supply-chain compromise, and potentially physical sabotage. Connectivity routes, power systems, and logistics corridors are all potential attack surfaces. Security therefore looks like critical infrastructure security: layered physical security, rigorous cyber segmentation between IT and industrial controls, strict access control, continuous monitoring, and adversary-aware resilience planning. Sovereignty matters because it shapes who governs the asset, what legal tools exist, and how it is defended. Under the U.S.-state assumption, Greenland becomes part of a national critical infrastructure frame, which can strengthen security posture and clarify response authorities.

On Mars, sovereignty is conceptually simpler and operationally stranger. There is no ordinary adversarial environment in the conventional sense—no local criminal ecosystem, no nearby rival state with easy physical access. But the system is still vulnerable, and in some ways more so, because it is fragile and isolated. Cybersecurity is still essential because a compromised control system could be mission-threatening. Supply-chain integrity is even more critical because you cannot quickly replace compromised components. "Security" on Mars therefore collapses into assurance: formal verification where possible, extreme access control, minimal external dependencies, and disciplined change management. The controls are less about defending against daily intrusion attempts and more about preventing catastrophic compromise in a system where recovery options are limited.

So Greenland's security posture emphasizes defense against active adversaries and hardening of infrastructure interfaces. Mars emphasizes assurance, minimalism, and catastrophic-risk avoidance in a high-fragility environment. Different controls for different threats.

7.5 Cost structure: capex, opex, and "distance tax"

The easiest way to misunderstand extreme infrastructure is to misread costs. The relevant concept is the "distance tax": costs that scale nonlinearly with remoteness because remoteness amplifies uncertainty and extends repair loops.

Greenland's capex is high because the site is remote, weather-constrained, and infrastructure-light. You pay more for construction logistics, for overbuilt redundancy, for inventory depth, and for specialized staffing and rotation. But Greenland's opex can be attractive under the thought experiment if nuclear baseload is stable and if cooling overhead is low. Over time, a well-run Greenland campus can convert high capex into competitive unit economics by achieving high utilization with low thermal overhead and stable energy costs.

The distance tax in Greenland is real but bounded. You can ship large components. You can schedule maintenance seasons. You can fly in expertise. You can gradually thicken local capability. The uncertainty can be managed with planning, modularity, and redundancy.

Mars is the distance tax in pure form. Capex is not just "higher"; it is a different category because every kilogram, every tool, every spare, and every redundancy layer has launch and mission costs. Opex includes not only energy and maintenance but the operational cost of isolation: long planning horizons, slow iteration, and heavy assurance processes. The opportunity cost of downtime is different too, because the mission's objectives may be time-sensitive and because the ability to recover quickly is structurally limited.

The economic conclusion follows from the constraint geometry: Greenland can plausibly amortize high capex into competitive ongoing cost per unit of AI output if the facility is designed for high utilization and manageable repair loops. Mars cannot plausibly compete as a commercial AI campus under near-term regimes because its distance tax is not a surcharge; it is the dominant term.

Summary of the matrix

If you compress the comparison to one sentence, it is this: Greenland is a remote industrial project whose constraints can be thickened into reliability with money, engineering, and governance; Mars is a mission environment where constraints compress the design space and force autonomy, scarcity, and survivability as first principles. That is why Greenland is far easier, and why "easier" is still not "easy."

8. Program Plan for a Greenland Pre-Trained AI Campus

A Greenland compute territory becomes credible only when it stops sounding like a single heroic build and starts looking like a program: phased milestones, reversible commitments, measurable outcomes, and a disciplined path from "pilot that works" to "campus that scales." The program plan below is intentionally optimized for pre-trained AI workloads. That choice is not cosmetic. It shapes everything: the network strategy, the artifact movement strategy, the staffing model, and the definition of success. The program treats nuclear baseload as available

by assumption, and it treats LEO connectivity as optimized enough to run a robust control plane and a bandwidth-limited service plane.

The overarching principle is **staged irreversibility**. Each phase should deliver real value on its own, while building the infrastructure primitives required for the next phase. A campus that tries to "skip to scale" in a harsh environment usually fails not because the destination is impossible, but because the learning loops are too slow and the first inevitable surprises arrive after the build is already overcommitted.

8.1 Phase 0: 5–20 MW pilot inference site

Phase 0 is not a "toy." It is the minimum meaningful unit of Arctic compute operations: large enough to stress systems, small enough to repair quickly, and valuable enough to justify continuous use.

The goals of Phase 0 are operational proof, not maximum throughput.

Site and physical build

The Phase 0 facility should be modular and fast to commission. Favor prefabricated electrical and mechanical rooms, containerized data halls, and a building envelope designed for wind, icing, and extreme thermal cycling. The construction philosophy is "factory-built, field-connected": reduce onsite complexity and maximize repeatability.

Power and electrical architecture

Treat Phase 0 as an islandable microgrid from day one. Even if you have grid interconnect, you design the campus to be able to operate independently and to restart independently. The electrical design should emphasize clean power delivery and ride-through: UPS strategy, staged load ramp, segmentation, and the ability to shed noncritical load automatically without losing control plane integrity.

Cooling and thermal posture

Phase 0 should demonstrate two cooling modes in practice: cold-air economization when conditions allow and liquid cooling for high-density racks. The purpose is to learn the Arctic edge cases: snow intrusion, icing, sensor drift, condensation behavior, and the operational realities of leak detection and coolant management. You want telemetry-rich truth, not marketing claims.

Connectivity posture

Use LEO as primary connectivity if desired, but treat control plane availability as sacred. Control-plane traffic gets priority, strong routing discipline, and clear degraded modes. Phase 0 should prove that the campus remains manageable under partial outages: "control-plane only," "service degraded," and "short isolation" modes.

Operations and staffing
Phase 0 builds the initial human system. The target is a small, highly trained local core team capable of first response, plus a remote operations center that can intervene interactively. The pilot is where you build the playbooks: incident taxonomy, change control, maintenance routines, spare replacement procedures, and escalation thresholds.

Deliverables and exit criteria
Phase 0 succeeds when it demonstrates sustained utilization and disciplined operations under real weather, real faults, and real maintenance. The exit criteria are not "we turned it on." The exit criteria are:

- predictable uptime with known failure patterns
- validated black-start capability
- proven degraded-mode operations
- stable remote ops tooling and governance
- a credible inventory strategy for high-failure components

If Phase 0 cannot be operated with confidence, Phase 1 should not be attempted. This is the first "hard gate" that prevents narrative from outrunning feasibility.

8.2 Phase 1: 50–100 MW campus with mature remote ops

Phase 1 is where the program becomes a campus rather than a pilot. At this scale, the facility stops being "a special project" and starts behaving like critical infrastructure with real economic output and real strategic relevance.

The goal of Phase 1 is to move from artisanal reliability to industrial reliability.

Campus architecture
Phase 1 should be built in blocks: multiple halls, each with its own fault domains and the ability to be isolated without taking down the entire site. Avoid a monolithic design where one failure can cascade. The campus should be segmented electrically and thermally, with clear containment boundaries.

Power scaling
In Phase 1, power strategy transitions from "pilot supply" to "industrial baseload." The design must provide both capacity and the stability required for dense GPU loads. The power system should be engineered for maintenance without downtime: redundant transformers, redundant distribution paths, and the ability to take major components offline while sustaining operations.

Mature remote ops
This is where remote operations becomes a product. Mature remote ops includes:

- high-fidelity telemetry with clear fault classification
- a disciplined change pipeline (canary, staged rollout, rollback discipline)
- automated incident responses for known classes of faults
- clear human escalation ladders with roles and rehearsal
- continuous configuration drift detection and correction

The campus should be operable with a small local team and a robust remote team. The human system is a key reliability primitive, not an afterthought.

Connectivity strategy
Phase 1 can still run with LEO as a major component, but the network architecture must be explicitly multi-path and resilience-first. Even if fiber is not yet present, the campus should use multiple gateways, routing diversity, and multi-constellation backup where possible. The goal is not maximum bandwidth; the goal is predictable operational behavior under stress.

Service posture
By Phase 1, the campus should support multi-tenant workloads: internal workloads (evaluation, distillation, archiving) and external workloads (inference services and scheduled fine-tuning). This requires governance: admission control, workload isolation, quota management, and security segmentation that treats the facility as a strategic shared resource rather than a single-purpose cluster.

Phase 1 exit criteria

- high utilization with stable unit economics
- demonstrated maintenance without major downtime
- proven incident response under real faults
- robust “degraded but useful” modes during connectivity instability
- a clear path to heat reuse integration and local value capture

8.3 Phase 2: 200+ MW expansion and heat-reuse integration
Phase 2 is where Greenland becomes a true compute territory. At 200+ MW scale, the campus is no longer a regional facility. It is a strategic industrial asset. That changes the engineering priorities: failure containment must be stronger, security posture must be more stringent, and community integration becomes critical for political stability.

The goal of Phase 2 is sustained, export-class throughput with local value capture.

Power and microgrid evolution
At this scale, the microgrid becomes a full utility-grade system. The campus must manage its own stability, reserve margins, and black-start logic at scale. It must be designed to ride through faults without cascading losses. Load shedding and power-aware scheduling become standard practice: the compute scheduler must respond to power conditions automatically.

Heat reuse as an engineered interface
Phase 2 is where heat reuse moves from "nice idea" to infrastructure. The campus should export heat via a deliberate interface: heat exchangers, buffer tanks, metering, and contracted delivery. Heat reuse should be stable and predictable even under varying compute loads. This requires thermal buffering and control logic that treats district heat as a service with uptime commitments.

Heat reuse can become the anchor of local legitimacy: reducing heating costs, supporting greenhouses or aquaculture, and turning the campus into a contributor rather than a consumer.

Campus modularity and failure domains
At 200+ MW, you must assume that something is always failing somewhere. Reliability is no longer about preventing failure; it is about preventing failure from becoming systemic. Phase 2 requires deeper segmentation: multiple independent halls, multiple independent electrical zones, and the ability to isolate faults quickly. The campus must be designed to lose a hall without losing the site.

Security and governance upgrades
A 200+ MW campus is an intelligence target. Security must be treated as critical infrastructure: strict physical security, strong cyber segmentation, and governance frameworks that limit who can run what workloads and where artifacts can move. "Sovereign compute" becomes meaningful: policy controls, auditability, and provable isolation boundaries.

Phase 2 exit criteria

- predictable high utilization with robust fault containment
- credible heat export with measurable local benefit
- hardened security posture consistent with strategic infrastructure
- demonstrated ability to scale without losing operational discipline

8.4 Workload strategy: inference, distillation, fine-tuning, eval, archiving
A Greenland campus works best when its workloads are aligned with its strengths: stable baseload power, cold-climate thermal advantage, and connectivity that is good enough for

control-plane and moderate service-plane traffic but not designed to be a perpetual bulk-ingest firehose.

The workload stack should therefore emphasize compute-heavy, transfer-light activities.

Inference
Inference is the primary revenue engine and the most natural fit. It benefits from stable power and high utilization. Many inference services are tolerant of modest network variance because generation time dominates. The campus can host multiple model families and route requests based on cost/latency targets.

Distillation
Distillation turns large models into smaller, cheaper ones. It is compute-heavy and can be run in batch. It also produces compact outputs—new weights and evaluation reports—making it network-friendly. Distillation is a natural "factory workload" for a remote campus: run long jobs, export distilled artifacts.

Fine-tuning
Fine-tuning for specialized domains is less bandwidth-intensive than frontier pretraining and can be structured around curated datasets shipped as job bundles. Parameter-efficient methods produce compact update artifacts, further reducing network load.

Evaluation
Evaluation is the governance backbone. A campus that runs inference at scale should also run continuous evaluation: regression tests, safety suites, performance characterization, and "canary comparisons" against known baselines. Evaluation is ideally batchable and produces structured results—perfect for a campus that can keep working even when connectivity is imperfect.

Archiving and artifact stewardship
A compute territory should treat models and their provenance as critical assets: versioned weights, evaluation logs, deployment manifests, and reproducible configurations. Archiving is a strategic function because it supports rollback, audit, and long-horizon reproducibility. It also makes the campus a durable node in a broader AI ecosystem, not just a transient compute source.

The unifying strategy is to **stage models locally, cache job bundles locally, compute locally, and export compact artifacts**. This is how you make LEO-compatible service patterns without pretending satellite is fiber.

8.5 Measuring success: uptime, cost per token, and resilience drills

Extreme infrastructure fails when success is defined as “it runs.” A compute territory must define success as stable, repeatable, auditable performance over time.

Three metric families matter most.

Uptime and service continuity

Uptime must be measured meaningfully: not just “the building had power,” but “the inference service met SLAs,” “the evaluation pipeline completed on schedule,” and “the control plane remained available during incidents.” Track downtime by root cause and by failure domain. The goal is not zero incidents; it is contained incidents and fast recovery.

Cost per token and utilization discipline

A pre-trained AI campus must translate its advantages—cooling, baseload power, high utilization—into stable unit economics. Cost per token is the simplest public-facing metric, but internally you also need utilization, energy overhead, and “idle-accelerator hours” as a measure of opportunity cost. The program should treat utilization as a reliability metric: high downtime masquerades as low utilization.

Resilience drills and operational truth

Resilience is not what you claim; it is what you can demonstrate on demand. The campus should run regular drills:

- simulated connectivity loss (control-plane only mode)
- partial power events and controlled load shedding
- black-start exercises
- hall isolation tests
- failover routing validations
- disaster recovery rehearsals for critical data

Drills should be treated as production events with logging and postmortems. The goal is to keep the organization honest: a remote campus can drift into complacency because disasters are rare—until they are not.

If these metrics are met across phases, Greenland becomes a credible compute territory for pre-trained AI workloads: high utilization, stable outputs, robust operations, and visible local value through heat reuse. And the comparison with Mars remains clear: Greenland’s constraints can be managed through phased engineering and disciplined operations, while Mars requires

autonomy-first, scarcity-first design that puts it in a fundamentally different category of infrastructure.

9. What This Thought Experiment Teaches About Earth-Based Compute Futures

The Greenland-versus-Mars comparison is deliberately exaggerated, but the exaggeration is the point. It functions like a stress test: by pushing the siting question into extreme cases, it reveals which variables remain decisive after you remove political noise and marketing slogans. The enduring lessons are not "move to the Arctic" or "don't go to Mars." The enduring lessons are about how AI compute is changing what counts as infrastructure, what counts as sovereignty, and what counts as feasible engineering when you scale beyond familiar assumptions.

The deepest takeaway is that AI compute is migrating toward the status of a strategic industrial base. That shift changes how we think about siting, redundancy, and governance even for mundane Earth locations. "Cold climates" are just one expression of a broader trend: compute is becoming heavy industry again—electricity-hungry, supply-chain-dependent, and tightly coupled to geopolitics. The thought experiment forces a clear-eyed view of that transition.

9.1 AI infrastructure as a new kind of strategic industrial base

For decades, digital infrastructure was framed as weightless: software, services, clouds, and networks that seemed to float above geography. AI punctures that illusion. At scale, AI compute is anchored in physical realities—power generation, heat rejection, specialized hardware supply chains, and resilient networks. That anchoring pushes AI infrastructure into the conceptual category once reserved for steel, petrochemicals, ports, and aerospace.

When an industry becomes "strategic," three things happen.

First, it becomes capacity-constrained in the real world. You can have infinite demand for tokens, but you cannot have infinite power delivery or infinite accelerator supply. The scarce resource becomes not ideas but physical throughput. States, firms, and coalitions respond by trying to secure capacity—through siting, procurement, subsidies, and sometimes protectionism.

Second, it becomes security-sensitive. The infrastructure that produces AI outputs becomes a target: for espionage, disruption, influence, and supply-chain compromise. The line between "IT asset" and "critical infrastructure" blurs. Governance shifts accordingly: stricter controls, more audit requirements, higher baseline security posture, and stronger assumptions about adversarial interest.

Third, it becomes geographically expressive. Once power and cooling dominate cost and feasibility, geography matters again. The old promise of "run your business from anywhere" is

replaced by a new truth: you can run AI services from many places, but you can only run them at extreme scale where power, thermals, and logistics allow. Geography re-enters the story not as a romantic variable, but as a binding constraint.

The Greenland thought experiment dramatizes this shift. It treats compute not as a laptop-and-internet project but as an industrial campus with a power plant attached. That is what "strategic industrial base" looks like in the AI era: a fusion of energy infrastructure and high-value computation, governed like a national asset rather than like a commodity service.

9.2 "Compute colonies" on Earth: the minimum viable primitives

"Compute colony" is a provocative phrase, but it captures a practical idea: a self-contained compute site designed to operate reliably under resource constraints and partial isolation, while still exporting value to the wider world. You do not need to be on Mars to behave like a colony. A remote Arctic campus, an island site, a desert facility, or even a hardened inland cluster can all take on colonial characteristics when they prioritize autonomy and resilience.

This thought experiment suggests a set of minimum viable primitives for any Earth-based compute colony.

Baseload power with reserve margins

A compute colony needs not just enough power but power that is stable, expandable, and resilient. Baseload matters because high utilization is the path to economic viability. Reserve margins matter because failure containment requires slack: you cannot survive faults if you are operating at the edge of capacity.

Thermal strategy that survives edge cases

Cooling must be designed for the worst days, not the average day. In harsh environments, failure modes often come from uncommon boundary conditions: storms, icing, heat waves, smoke, dust, or unexpected humidity. A colony must be able to degrade gracefully without hardware damage.

A microgrid with islanding and black start

Whether or not the site connects to a broader grid, it must be able to operate independently and recover independently. This is not a fetish for autonomy; it is a practical requirement when you cannot assume external stability. Islanding and black start are the difference between "a fault is a hiccup" and "a fault is a multi-day outage."

A control plane that survives bandwidth collapse

The control plane must remain alive even when the service plane is impaired. This means traffic prioritization, out-of-band access, strict segmentation, and explicit degraded modes. A compute colony dies when it becomes unobservable.

Inventory depth and repair loop design
Remote sites win or lose on mean time to repair. That means modular replacement, on-site spares for known high-failure components, and rehearsed procedures. Logistics is not external to reliability; it is part of reliability.

A workforce model that is stable under rotation
A compute colony needs a human system that does not collapse when one person leaves. That means training pipelines, standardized procedures, remote expert support, and a local core that can respond when the network is down and the weather is bad.

Security posture consistent with strategic infrastructure
As sites become more valuable, their threat models intensify. A colony must assume adversarial interest and design for defense, detection, and recovery.

These primitives generalize. They apply not only to Greenland but to any future where compute becomes geographically distributed, power-hungry, and strategically significant. The "minimum viable colony" is an emerging pattern of infrastructure design.

9.3 The boundary between engineering feasibility and narrative temptation
The modern technology landscape is crowded with narratives that sound like engineering. "Cold places are perfect." "Satellites solve connectivity." "Nuclear solves power." "Mars is just a harder version of Earth." These narratives often contain a grain of truth, which makes them dangerous: they encourage people to skip the coupled-system analysis and leap to slogans.

The thought experiment draws a boundary between what engineering can do and what narrative tries to smuggle in.

Engineering feasibility is characterized by measurable primitives, testable phases, and repair loops that close inside acceptable time horizons. It is honest about failure modes and builds graceful degradation. It treats operations as a first-order system.

Narrative temptation is characterized by single-variable thinking, implicit magical assumptions, and the quiet erasure of repair loops. It treats a successful demo as a proof of scalable reality. It assumes that because a thing is possible in principle, it is feasible in practice.

Greenland sits near the boundary. With permissive assumptions, it looks feasible—hard but feasible—because the repair loop can close and because the operational tempo can remain terrestrial. Mars sits on the other side of the boundary for any data-center-like interpretation of AI compute, because the latency and resupply constraints stretch repair loops beyond ordinary operational time scales. Mars demands a different kind of system: autonomy-first, scarcity-first, survival-first.

The key lesson for Earth-based compute is that feasibility is not a function of ambition. It is a function of whether your failure modes can be contained and your repair loops can be closed. If they cannot, you are doing narrative, not engineering.

9.4 A reusable framework for siting decisions in extreme environments

The point of the Greenland–Mars comparison is to extract a reusable decision framework that can be applied to real siting problems on Earth: Arctic regions, deserts, islands, conflict-adjacent zones, and energy-rich but infrastructure-poor territories.

A reusable framework begins with a discipline of questions. The questions should be asked in an order that forces reality to appear early.

Step 1: Identify the binding constraint class

Is your project power-limited, network-limited, logistics-limited, or governance-limited? Most projects have many constraints; the question is which one binds first as you scale.

Step 2: Define the workload geometry

Are you doing frontier pretraining, pre-trained inference, batch evaluation, or hybrid? Workload determines network needs, storage patterns, operational sensitivity, and value density. If you pick the wrong workload for the site, no amount of optimization saves you.

Step 3: Specify the reliability model, not just the architecture

What are the expected failure modes? What are the repair loops? What is the on-site inventory depth? What is the mean time to repair under realistic conditions? If you cannot answer these, the site is not defined yet.

Step 4: Design degraded modes as first-class behaviors

Assume connectivity loss. Assume partial power events. Assume thermal stress. Specify what the system does in each case. If the degraded modes are undefined, you are assuming miracles.

Step 5: Compute the "distance tax" explicitly

How does remoteness amplify cost and uncertainty? What is the cost of a delayed repair? What is the cost of a missed shipping window? What is the cost of a specialist being unavailable? Distance tax is not linear; it often appears as nonlinear risk.

Step 6: Phase the program to preserve learning loops

Design Phase 0 to reveal operational truths quickly. Design Phase 1 to industrialize reliability. Design Phase 2 to scale without losing discipline. Never skip phases in harsh environments unless you want your first surprise to arrive after irreversibility.

Step 7: Integrate local value capture and legitimacy

A compute colony survives politically when it provides durable local benefit: heat reuse, jobs,

training, infrastructure upgrades, research partnerships. If the site is perceived as extractive, it becomes fragile regardless of engineering excellence.

This framework can be used for Greenland, but it also scales down to more mundane choices: where to place new AI capacity in power-constrained regions, how to decide between grid-tied and islandable designs, how to evaluate satellite versus fiber, and how to match workload types to infrastructure realities.

The thought experiment's final lesson is methodological: siting is not a map problem. It is a coupled-systems problem. The winning site is the one whose constraint geometry matches the workload, whose repair loops close, whose degraded modes are designed, and whose social legitimacy is earned. That is the future of Earth-based compute: not merely bigger clusters, but more deliberate, colony-like infrastructure that treats power, resilience, and governance as first-class engineering objects.

10. Conclusion

The Greenland-versus-Mars comparison works because it is unfair. It uses extremes to expose the governing constraints that polite, incremental siting debates often obscure. Under the thought experiment—Greenland becomes a U.S. state, nuclear friction is removed, and LEO connectivity is adapted—Greenland can plausibly host a serious pre-trained AI campus. Mars cannot, at least not in anything resembling the terrestrial data center sense. But the most important contribution of this paper is not the ranking. It is the method: a way to reason about compute infrastructure as a coupled system of power, thermals, networking, repair loops, and governance—where feasibility is shaped less by ambition and more by which constraints bind first.

10.1 Greenland is easier than Mars, and why that's the wrong level of claim

"Greenland is easier than Mars" is true in the same way that "a storm is easier than space" is true. It is correct but not yet useful. It tempts readers into thinking the argument has been made when it has only been framed.

The wrongness of the claim is not factual; it is categorical. The comparison is not between two sites on a common scale. It is between two different classes of infrastructure problems. Greenland is a remote industrial build that still lives within Earth's envelope: materials behave predictably, human labor is unconstrained by life support, repair loops can close on human time scales, and logistics—while difficult—remain continuous. Mars is a mission environment where the control loop is delayed by physics, resupply is episodic, human labor is bounded by habitat constraints, and power is rationed in a settlement economy. Once you say "Mars," you are no longer doing siting; you are doing civilization engineering.

So the correct conclusion is not “choose Greenland.” The correct conclusion is that Greenland belongs to the class of projects where iterative engineering and disciplined operations can plausibly widen the feasible region. Mars belongs to a class where feasibility is narrow unless the underlying infrastructure basis changes dramatically.

10.2 The deeper result: feasibility follows constraint geometry

The central result of the paper is methodological: feasibility follows constraint geometry. That is, the viability of a compute site is determined by the interaction of constraints, not by any single advantage.

Cooling advantage does not compensate for weak repair loops.
Cheap power does not compensate for operational latency that prevents timely intervention.
Connectivity does not compensate for power scarcity.
Regulatory permissiveness does not compensate for logistics physics.

The Greenland thought experiment becomes plausible only because the constraints align: baseload power removes the dominant bottleneck, cold climate reduces thermal overhead, LEO connectivity supports an interactive control plane and moderate service plane, and terrestrial logistics allows repair loops to close. The workload choice—pre-trained inference and batchable post-training tasks—further aligns the site with its constraints by reducing dependence on continuous bulk ingest.

The Mars case fails not because one variable is missing, but because the constraint geometry is hostile across multiple dimensions simultaneously: minutes of latency, kilowatt-scale power regimes, engineered thermal rejection under dust and thin atmosphere, and maintenance under habitat constraints. Each of these would be challenging alone. Together they compress the feasible region until “commercial AI campus” ceases to be the right mental model.

This is why the paper’s framework generalizes. It is not about Greenland. It is about how to think when compute becomes heavy industry: identify the binding constraints, match workloads to infrastructure realities, design degraded modes explicitly, and ensure that repair loops can close at the tempo your system requires.

10.3 What would have to change for Mars to become competitive

Mars could host meaningful computation, and will, but for Mars to become competitive with Earth-based compute for anything resembling large-scale AI services, several structural shifts would have to occur. These are not incremental improvements; they are regime changes.

Power must move from scarcity to industrial baseload

Mars compute becomes “campus-like” only if power supply moves from kilowatt-class settlement budgets to sustained megawatt-class (and beyond) baseload with redundancy. That

implies not just a reactor or a solar farm, but an entire electrical ecosystem: generation, storage, distribution, protection, and maintenance capacity scaled for industry, not survival.

Operations must be autonomy-native at the infrastructure level
Mars must treat autonomy as the default operating system: self-diagnosis, self-repair where possible, local policy enforcement, and safe-state transitions that do not require real-time human intervention. This is deeper than "add automation." It requires infrastructure designed for autonomy as a first principle.

Maintenance must be reimagined around robotics, modularity, and indoor serviceability
A Mars compute facility must minimize the need for human EVA maintenance and make most service tasks executable in controlled environments. That implies a design that looks more like a factory line than like a data center aisle: modular swap units, robotic handling, internal service corridors, and a spares economy planned as part of the settlement's industrial base.

Thermal rejection must become robust and dust-tolerant
Mars cooling cannot be an extrapolation of terrestrial cooling. Heat rejection must be engineered with radiators and dust management as primary design constraints. A "commercial" facility demands thermal reliability comparable to Earth; that is a tall order under Martian conditions unless the thermal system is radically overdesigned.

Logistics must become frequent, cheap, and high-capacity
Even with improved autonomy, a large compute operation will consume parts, tools, and replacement modules. Mars becomes competitive only if resupply becomes routine rather than episodic—meaning transport systems that reduce the distance tax by orders of magnitude.

Even if all of these change, the speed-of-light delay remains. Mars will always be latency-separated from Earth. That means the product profile would still differ: Mars compute would likely serve Mars-first needs (settlement autonomy, science, local manufacturing optimization, closed-loop life support control) rather than Earth-first interactive services. "Competitive" might mean "valuable and indispensable locally," not "cheaper per token than Greenland."

10.4 Forward work: rigorous scenarios, benchmarks, and a disciplined Reference section
The next step is to turn the thought experiment into a disciplined research program rather than a rhetorical comparison. The forward work falls into three categories.

Rigorous scenarios
Define multiple scenario sets that vary the assumptions rather than treating them as fixed:

- Nuclear baseload available vs constrained
- LEO-only connectivity vs LEO + fiber

- Inference-dominant workloads vs mixed workloads with some training
- Local heat reuse integrated vs not integrated
- Different threat models and governance regimes

Each scenario should specify a concrete facility scale (MW), a concrete workload mix, and explicit degraded modes.

Benchmarks that map to the real constraints
Move beyond toy metrics and specify benchmarks that align with the constraint geometry:

- uptime under defined fault injection (connectivity loss, partial power events)
- mean time to repair for representative failures under realistic logistics constraints
- cost per token under defined utilization targets
- resilience drill performance (black start, hall isolation, control-plane survival)
- artifact movement efficiency (model deltas, eval bundles, audit logs)

These benchmarks are not academic; they are the language of feasibility.

A disciplined Reference section
Finally, the paper needs a Reference section that does not merely decorate the argument but anchors it. The references should be chosen to support the core claims: latency regimes, power regimes, nuclear deployment classes, microgrid design principles, data center thermal engineering in extreme environments, satellite network performance characteristics, and deep-space operations doctrine. The goal is not to overwhelm the reader with citations but to make the inference chain auditable: when the paper says “this is a hard limit,” the reader should be able to see what that limit is grounded in.

If that forward work is done, the paper becomes more than a thought experiment. It becomes a template for decision-making in an era where compute is heavy industry, geography is back, and “where to put the machines” is inseparable from power, resilience, and governance. Greenland and Mars are the extremes that make the pattern visible. The pattern is what matters.

References

R1. Greenland institutions, buildability, and local context

R1.1 Statistics Greenland. *Greenland in Figures 2024*. Stat Greenland

R1.2 Ilisimatusarfik (University of Greenland). *Artificial intelligence in Greenland* (event listing). Ilisimatusarfik

R1.3 University of the Arctic (UArctic). *PhD course on new technologies and local perspectives in Arctic health research at Ilisimatusarfik completed* (news post). UArctic

R1.4 Ilisimatusarfik (University of Greenland). Official site / overview. Ilisimatusarfik

R1.5 ARTEK (Arctic Technology Centre, Sisimiut; DTU collaboration). Official overview. DTU

R1.6 Kalaallit Airports. Greenland airport development program (official information). uni.gl

R2. Data center energy, siting, and efficiency fundamentals

R2.1 Shehabi, A. et al. *United States Data Center Energy Usage Report* (Lawrence Berkeley National Laboratory).

R2.2 U.S. Department of Energy (FEMP) / NREL. *Best Practices Guide for Energy-Efficient Data Center Design* (July 2024). The Department of Energy's Energy.gov

R2.3 International Energy Agency (IEA). *Electricity 2024* (data centers and networks as a demand driver).

R2.4 The Green Grid. *The Green Grid Data Center Power Efficiency Metrics: PUE and DCiE* (white paper). datacenters.lbl.gov

R2.5 ISO/IEC. *ISO/IEC 30134-2: Data centres—Key performance indicators—Part 2: Power Usage Effectiveness (PUE)*. ISO

R2.6 ASHRAE. *ANSI/ASHRAE Standard 90.4 (Energy Standard for Data Centers)* and addenda (example addendum). ASHRAE

R3. Cooling in cold climates and liquid cooling at high density

R3.1 ASHRAE. *Thermal Guidelines for Data Processing Environments* (Datacom Series; referenced by DOE guide). The Department of Energy's Energy.gov

R3.2 Open Compute Project (OCP). Liquid cooling / facility guidance (direct-to-chip and related specs). Open Compute Project

R3.3 Microsoft. *Project Natick* (underwater data center research and reliability outcomes). Source

R3.4 U.S. DOE / NREL. Guidance on economization, airflow management, and facility-side efficiency tradeoffs. The Department of Energy's Energy.gov

R4. Reliability, redundancy, and data center performance standards

R4.1 Uptime Institute. *Tier Standard: Topology* (classification basis). Uptime Institute

R4.2 Uptime Institute. *Annual Outage Analysis 2024* (executive summary PDF). Uptime Institute

R4.3 Telecommunications Industry Association (TIA). *ANSI/TIA-942 ratings / certifications*

overview. TIA Online
R4.4 TÜV NORD. EN 50600 overview (European data center standard framing). TÜV NORD

R5. Heat as an output: waste-heat reuse and district heating integration

R5.1 Fortum. *Fortum and Microsoft's datacentre project* (district heating integration and scale claims). Fortum
R5.2 Stockholm Data Parks. Program overview / heat reuse model. ScienceDirect
R5.3 Review literature on data center waste heat recovery and district heating integration (example review). ScienceDirect

R6. Nuclear baseload for data centers: microreactors, SMRs, and "no-friction" governance contrast

R6.1 U.S. Department of Energy. Microreactor program overview (deployment rationale, power ranges). cto.mil
R6.2 U.S. Department of Defense / Project Pele program background (microreactor for installations). DTU
R6.3 International Atomic Energy Agency (IAEA). *Small Modular Reactors: Advances in SMR Developments 2024* (ARIS update). IAEA+1
R6.4 U.S. Nuclear Regulatory Commission (NRC). *Part 53 rulemaking* (risk-informed, technology-inclusive framework status). Nuclear Regulatory Commission+1
R6.5 Federal Register. *Risk-Informed, Technology-Inclusive Regulatory Framework for Advanced Reactors* (Part 53 proposed rule notices). Federal Register+1
R6.6 NRC. *10 CFR Part 52* (combined license framework; backgrounder / eCFR). Nuclear Regulatory Commission+2Nuclear Regulatory Commission+2

R7. Microgrids, islanding, black start, and power quality for GPU campuses

R7.1 NREL. Microgrid design / resilience guidance (islanding, controls, integration patterns). ESA Blog
R7.2 IEEE. *IEEE 1547-2018* (DER interconnection and interoperability baseline for grid-tied resources). Light Reading
R7.3 Power quality engineering references for harmonics and sensitive loads (data center context). c03.apogee.net+1

R8. LEO connectivity as control plane and product surface

R8.1 Starlink (SpaceX). *Specifications* (latency, performance characteristics). Open Compute Project+1
R8.2 Ookla. Reporting on Starlink performance and latency trends (methodology and measurement framing). IEA

R9. Mars operations: when physics is the policy

R9.1 NASA. *Mars Facts / Mars overview materials* (environmental constraints baseline). IEA

R9.2 NASA Human Factors / communications-delay framing (cognitive and operational implications). IETF Datatracker

R9.3 NASA NTRS. *Human Exploration of Mars Design Reference Architecture 5.0 (DRA 5.0)* (summary record). NASA Technical Reports Server

R9.4 NASA. *Mars DRA 5.0 executive summary presentation* (program-level architecture framing). nasa.gov

R9.5 NASA/SP addendum materials for DRA 5.0 (architecture detail extension). NSS

R9.6 NASA/JPL. Deep Space Network overview (interplanetary comm infrastructure context). nasa.gov

R9.7 JPL. *Deep Space Network Services Catalog* (TT&C services and operational constraints). NASA

R9.8 Mars thermal control / radiator and dust-deposition considerations (example technical source). cto.mil

R10. Delay-tolerant networking for high-latency, disruption-prone regimes

R10.1 IETF. *RFC 4838: Delay-Tolerant Networking Architecture* (bibliographic record). IETF Datatracker

R10.2 IETF Datatracker. DTN architecture refresh draft (4838bis workstream). IETF Datatracker

R10.3 APNIC. DTN performance discussion (operational perspective, store-carry-forward intuition). APNIC Blog

R11. AI infrastructure "campus reality": vendor reference architectures and density assumptions

R11.1 NVIDIA. *DGX SuperPOD Reference Architecture (GB200 era)* (power-density, facility requirements, availability targets). NVIDIA Docs

R12. Cybersecurity and operational governance (relevant to remote ops and critical infrastructure)

R12.1 NIST. *SP 800-53 Rev. 5: Security and Privacy Controls for Information Systems and Organizations*. NIST Publications

R12.2 NIST. *Cybersecurity Framework (CSF) 2.0* (governance and risk framing update). NIST CSRC

The author has published over 4,800 AI-assisted papers at: https://www.researchgate.net/profile/Douglas-Youvan/research . Google Scholar has indexed these preprints, and if you want a particular reference, the AI mode of a Google search (Gemini) has efficiently catalogued these preprints.